There are different folders in the project folder:

1. **CrawlerDiffWays**: This folder contains 2 files, which we have tried to scrape the content of an article from the news sites.

Wefound step b, a better source of information for real time, reason being that many a times rss/xml feeds are older than a day, but it depends on different news sites.

1. **News\_streamer.py** –This file collects news\_feeds from various sites.

After collecting these feeds, each feed will be cleaned(removal of unnecessary text), tokenized and stemmed and stored in Elasticsearch, which is commented as of now(out of scope as of now)

self.news\_sources : all the URLs of different news websites, this can be stored in a file and then fetched. We can also setup a cron job to do the same..

server\_addr : ES server address - Will be localhost once elastic search is installed

All feeds are collected as RSS/XMLS's.

1. **Web crawler News.py –** This file contains article information from a specific news site, Business standard in our example. Every news site has a different site structure. Thus every site will have its own scraper which needs to be written.

1. **Crawlers** – Once we did the above test and wrote our own set of code. We were able to understand the scrapers that were written by the company. This folder has some sample code of how the scraper was actually written.

Also This folder has many subfolders explained below:

1. **Scripts** : It contains scripts which are called from a cronjob, which accumulates the data output(json) of all scrapers(in folder -> spiders).
2. **Spiders**: It has all individual news site scraper.
3. **Et.json**: Example of how the json looks like.
4. **Exploratory analysis** – This folder has an excel file, which contains summary level charts, which gives EDA.
5. 20170518\_eda\_summary\_charts: This file has all the summary level charts that we have created by analyzing news articles from **data.json,** which is one day’s worth of scraped articles data(Got it form the company)
6. **Data** – This folder contains, what we have extracted, from a **data.json(**this is not attached as it was a 2 GB file**)**file which we got from the company after 1 day’s job was run for all news sites. In addition to that we also got the below set of files.
7. **DataExtractJSON.py**: We were extracting the data from data.json and putting it to a data frame. So that sample files could be created with features which we want.
8. **Title\_and\_article1000.csv**: This is a sample file given by the company to use, so that we could run our content and collaborative based algorithms on it.
9. **TrainSet700\_User\_Ratings**: This is what we have created after giving ratings to each and every of 1000 sample articles. We further broke it down to 700 and 300 articles as train and testing samples respectively.
10. **User\_Ratings\_Compile**: This file is a conglomeration of all ratings given by 5 users. This is used for Collaborative recommendation technique.
11. **Recommendation –** This folder contains code for Content based and Collab based recommendation techniques.
12. **CollabTest.py:** Code written for Collaborative recommendation technique. This takes **User\_Ratings\_Compile.xlsx** as an input and gives out files for each of the users for 1000 articles.
13. **ContentTest.py:**  Code written for Content based recommendation technique. This takes **Title\_and\_article1000.csv**  as an input and gives out an output file, which has each article and 10 articles which are similar to it.
14. **OutputFolder\_Recommendations –** This contains all o/p data that we have got from the various recommendation code, present in Recommendation folder.
15. ContentOutput: This file is what we have got by running ContentTest.py code**.**
16. CollabOutput: This file is what we have got by running CollabTest.py code**.**
17. **Recommender\_Project\_Presentation v1-090717.pptx –** This is the ppt that has been created explaining the project.